

RESEARCH OUTPUTS / RÉSULTATS DE RECHERCHE

Multistream Proportional Fair Packet Scheduling Optimization in HSDPA

Vanderpypen, Joël; Schumacher, Laurent

Published in:

Proceedings of the 15th Annual Symposium of the IEEE/CVT Benelux Chapter SCVT 2008

Publication date:

2008

Document Version

Early version, also known as pre-print

[Link to publication](#)

Citation for pulished version (HARVARD):

Vanderpypen, J & Schumacher, L 2008, 'Multistream Proportional Fair Packet Scheduling Optimization in HSDPA', *Proceedings of the 15th Annual Symposium of the IEEE/CVT Benelux Chapter SCVT 2008*.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Multistream Proportional Fair Packet Scheduling Optimization in HSDPA

J. Vanderpypen and L. Schumacher, FUNDP – The University of Namur, {jva,lsc}@info.fundp.ac.be

Abstract—To improve wireless channel capacity, one can reduce radio resource waste through the scheduling of multiple users within a TTI. In interference-limited systems, user selection should be performed carefully, so as to limit interference. In this paper, we present a multistream scheduler, based on a Proportional Fair strategy, for its trade-off between fairness among users and cell throughput. This scheduler is a refinement of an earlier proposal, where the queue length of each user is taken into account, to improve the overall resource efficiency. To avoid significant computations, a heuristic has been developed. The scheduler performance is compared against a multistream Round Robin scheduler and four traditional single user schedulers via computer simulations, in the peculiar case of HSDPA transmissions. Both throughput and fairness gain are highlighted, especially with low users queue rates. A 8% gain in total cell throughput is achieved, while preserving fairness.

Index Terms—Multistream scheduling, Proportional Fair, HSDPA.

I. INTRODUCTION

In a cellular network, all users of a given cell are connected to the same base station, and the transmissions of all users are controlled by a scheduling algorithm. For each Time Transmit Interval (TTI), the scheduler shares the transmission resources, which are transmit power, spreading codes, frequency bands, LTE's Physical Resource Blocks (PRB) and so on. In interference-limited high speed cellular networks like HSDPA (High Speed Downlink Packet Access), the standard prescribes that a single user is scheduled in each cell for a TTI. Actually, it has been shown in [1] that this approach is suboptimal because when channel quality is poor, the scheduled user can not fully benefit from the allocated resource. So it would be more efficient to share the available resources among several users. As [2] mentions, the selection of these simultaneous users should be done carefully, to limit interference between each other.

There are many scheduling algorithms. The simplest one is the Round Robin (RR) scheduler, where each user is scheduled at a periodic time slot. This algorithm is really fair between users, but its throughput is not optimal. Actually, it is more efficient to schedule users when their variable channel is in good condition, i.e. when users would benefit the most from the resources they have been allocated. Another scheduling strategy, called Max Throughput, is therefore to schedule the user who has the best channel. The trouble with this technique is fairness among users: a distant user will never get scheduled as long as there are users with better channel conditions. There are

also some algorithms taking QoS into account, as the Max Weight scheduler for example. It can offer some guarantees on delay and delay jitter thanks to its criteria, which is the potential throughput weighted by the queue length. To achieve a good trade-off between fairness among users and cell throughput, we chose the Proportional Fair algorithm, where the criterium to maximize is also the potential throughput of the user, but normalized by its mean throughput so far.

This paper will detail the behaviour of HSDPA-enabled cellular networks. In HSDPA, there are 15 physical channels, determined by orthogonal channelisation codes, that are shared between active users. Selecting exhaustively the best group of simultaneous users, and the optimal code repartition implies heavy computations every TTI. Let us point out that in HSDPA, a TTI only lasts 2 ms [3]. This is the reason why we looked for a heuristic.

The algorithm we propose is based on a Proportional Fair criterium, where we originally introduce some QoS parameter. The algorithm is split into two phases: a first phase consists of identifying the best users to schedule, and the second one allocates them the radio resources.

The rest of the paper is organized as follows. Section II presents the system and the channel model. Our multistream scheduling algorithm is detailed in Section III, and its performance are evaluated within Section IV. Finally, Section V draws some conclusions about this work.

II. SYSTEM MODEL

Since our scheduling criteria is based on Proportional Fair, it requires knowledge on potential throughput and mean throughput. It is quite easy for the base station to log the throughputs of its users and evaluate their mean throughput. But as far as the potential throughput is concerned, it needs to be estimated, depending on resource allocation. So our first step is to get a rough estimate of the achievable bit rate of each user in the next TTI. This rate is upper bounded by the Shannon capacity [4]:

$$C_k = BW \cdot \log_2(1 + SNIR_k) \quad (1)$$

where the channel capacity of the user $k=1,2,\dots,N_{Queued}$ depends on the bandwidth BW and the Signal to Noise plus Interference Ratio $SNIR$ of its channel. These channel capacities are then truncated with respect to the queue length Q_k .

Hence, the potentially achievable throughput B_k is given by:

$$B_k = \min[C_k, Q_k / TTI] \quad (2)$$

¹J. Vanderpypen acknowledges the funding support of the "Fonds à la formation pour la Recherche dans l'Industrie et dans l'Agriculture" (F.R.I.A.).

This truncation is an original amendment to the algorithm proposed in [5]. The bound Q_k / TTI actually represents the bit rate which would empty the queue Q_k within the duration of one TTI. Indeed, even with a large channel capacity, if a user only has a short queue, it is worthless to over-estimate his/her throughput.

The received signal power S_k of a user k depends on the channel state $|H_k|$, his/her transmit power P_k , and his/her path loss d_k^γ , which represents the signal attenuation due to the distance d_k between the base station and the user k . The received signal power is obtained from

$$S_k = |H_k|^2 \frac{P_k}{d_k^\gamma} \quad (3)$$

The thermal noise is a constant. As far as interference are considered, because HSDPA is based on orthogonal codes, there is no interference between transmissions of a cell in the downlink as long as these transmissions are synchronized. However, because of multipath fading, several echoes of each transmitted signal will be received, and will bring interference into the system. The proportion of the transmitted signal which will be perceived as interference can be estimated by the orthogonality factor β [6]:

$$\beta = 1 - \frac{\sum_{i=1}^L (\alpha_i)^4}{\left(\sum_{i=1}^L (\alpha_i)^2 \right)^2} ; \quad (4)$$

where L is the number of taps modelling the channel dispersion, and (α_i) is the power of the i^{th} one. One can see that if there is only one significant echo, we have $\beta \approx 0$, so there is no interference.

As a result, the SNIR of a given user k can be estimated by

$$SNIR_k = \frac{|H_k|^2 \frac{P_k}{d_k^\gamma}}{\sigma_k + |H_k|^2 \frac{\beta_k P_{max}}{d_k^\gamma}} \quad (5)$$

where σ_k represents the noise power, and P_{max} the total emitting power of the cell. Once the SNIR is known for each user, all PF criteria can be obtained from (2), and scheduling choices can be done. These choices will be detailed within the next section.

III. SCHEDULING ALGORITHM

The criteria we will use to compare the users is based on Proportional Fair, to achieve a good trade-off between fairness among users and cell throughput. It consists of maximizing B_k , the potential throughput of the user k , weighted by \bar{B}_k , its mean throughput:

$$\frac{B_k}{\bar{B}_k} \quad (6)$$

So the scheduled user will have a high throughput thanks to B_k , and the fairness is achieved by the mean throughput weighting. For example, if a user has constantly a good channel and backlogged data to transmit, s/he will initially be frequently scheduled. But after some time, his/her mean throughput will increase, and the ratio in (6) will get lower than the one of another user with poorer channel quality which still has not transmitted yet.

To find the optimal resource allocation, we should evaluate this Proportional Fair criteria for all possible groups of scheduled users, for all possible code repartition between them. This 2-dimensional exhaustive search, in user and code spaces, can not be performed within a TTI. Only for the user selection, with 15 HSDPA codes, the total number of possible user groups is

$$\sum_{k=1}^{15} C_{N_{\text{queued}}}^k \quad (7)$$

And then for each group, we should test all code combinations. Hence, our interest for a heuristic.

In [5], it is suggested to first select users without taking into account the interference, so this selection can be performed linearly. When the scheduled users are known, the next step is to share the resource among them.

So our first step is to evaluate the potential throughput each user would obtain from one radio resource quantum, which can be a frequency band, a PRB, or a set of orthogonal codes, with the corresponding fraction of the transmit power. In our case of HSDPA transmissions, where there are maximum 15 codes to share and no power control, a quantum will be a single code, powered with 1/15 of the transmit power devoted to HSDPA.

Actually, with CDMA-based transmissions, like HSDPA, multi-user interference only depends on the transmit powers involved. Indeed, since all users use the same spreading factor, the interfering effect of a given code does not depend on the peculiar user whom the code belongs; multiple access interference (MAI) and intersymbol interference (ISI) are mistaken within each other

So if we consider that all 15 codes are used, we can estimate the throughput each user can get from one code, with 1/15 of the power P_{HSDPA} allocated to HSDPA transmissions. And this estimation can be done without any knowledge on the set of the other simultaneous users. Therefore, our second step is a loop on the resource: while there are still available codes, find the user which would maximize his/her PF criteria and allocate him/her the number of codes s/he needs. Obviously, we can not allocate a fraction of code. The number of codes are therefore rounded to the next integer towards zero, unless rounding leads to zero. A user should have at least one code to transmit, but we want to avoid wasting resource.

The multistream Proportional Fair scheduler can be synthesized in the following algorithm (I). To evaluate the multistream gain, we compared the new scheduler with a multistream Round Robin and four traditional mono-user schedulers: Round Robin, Max Throughput, Proportional Fair, and Max Weight.

1. Evaluate the throughput each user would benefit from a single code with 1/15 of the HSDPA transmit power:

$$\hat{B}_k = \min \left[BW \cdot \log_2 \left(1 + SNIR_k \right), Q_k / TTI \right], \quad \forall k=1, \dots, N_{Queued} \quad (8)$$

$$\text{where } SNIR_k = \frac{\left(\overline{H}_k \right)^2 \frac{P_{HSDPA}/15}{d_k^\gamma}}{\sigma_k + \left(\overline{H}_k \right)^2 \frac{\hat{B}_k P_{max}}{d_k^\gamma}} \quad (9)$$

2. While there are still available codes, find the user which would benefit the most from them:

1. For all user $k=1, \dots, N_{Queued}$:

- Evaluate the number of codes they need to maximize their throughput (i.e. empty their queue):

$$nc_k = \left\lceil \frac{Q_k}{\hat{B}_k \cdot TTI} \right\rceil - 1; \quad (10)$$

where $\lceil x \rceil$ is the smallest integer bigger than x .

- Truncate this number of codes by the number of codes still available;
- Evaluate the PF criteria of each user:

$$PF_k = \frac{\min \left[nc_k \cdot \hat{B}_k, Q_k / TTI \right]}{\hat{B}_k}; \quad (11)$$

2. Attribute $nc_{\tilde{k}}$ codes to the user $\tilde{k} = \arg \max_{k=1, \dots, N_{Queued}} PF_k$.

Algorithm (I) : Multistream Proportional Fair scheduling algorithm.

The Round Robin scheduler selects each user at a periodic time slot, and the multistream Round Robin also allocates resources in a carrousel kind of way, but if the selected user can not benefit from all codes, the unused ones can be allocated to the following users.

The Max Throughput chooses the user i_{MT} which has the maximum throughput:

$$i_{MT} = \arg \max_{k=1, \dots, K} B_k \quad (12)$$

The traditional Proportional Fair selects the user i_{PF} that maximizes

$$i_{PF} = \arg \max_{k=1, \dots, K} \frac{B_k}{B_k} \quad (13)$$

And finally, the Max Weight scheduler select the user i_{MW} that maximizes

$$i_{MW} = \arg \max_{k=1, \dots, K} B_k \cdot Q_k \quad (14)$$

As a matter of fair comparison, all these schedulers have relied on the bounded throughput (2). The next section will now evaluate the performance of the multistream scheduler.

IV. MATLAB SIMULATIONS

To evaluate the performance of the new scheduler, we did some computer simulations in MATLAB. We used SCME, a channel model implemented by the IST-WINNER project [7] according to the 3GPP standardization group specifications.

We considered 8 users uniformly distributed in a ring, between 35 and 500m away from the base station, moving with a velocity of 10 m/s. The bandwidth of the system is 5 MHz, and the carrier frequency is 2 GHz. As [8, Table 12.7] mentions, the maximum transmit power dedicated to HSDPA is 45 dBm, and the thermal noise power is -101.2 dBm.

To model the queue rates, we considered a fixed arrival packet rate of 1 packet per TTI, and a uniformly distributed packet size. With a packet size uniformly distributed in [1.5 kb, 4.5 kb], we get an average queue rate of 1,500 kbps per user, which means that the cell has to transmit for the 8 users in average 2,400 kb during the 100 TTIs of our simulations. We also considered twice and half of this queue rate.

Following figures compare the cumulative probability density function of the cell throughput for the different schedulers, with different mean queue rates. Results are averaged over 200 runs of 100 TTIs each. Fig. 1 considers a low mean queue rate of 750 kbps for each user, Fig. 2 1,500 kbps, and Fig. 3 3,000 kbps.

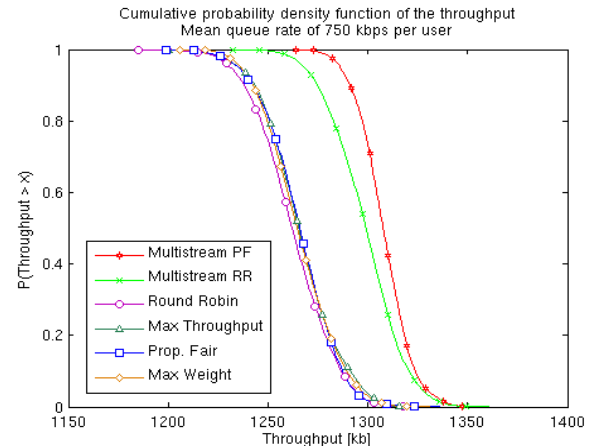


Fig. 1 : Cumulative probability density function of the cell throughput, for 8 users with a mean queue rate of 750 kbps each.

As one can see on Fig. 1, with such low queue rates, all traditional schedulers lead to the same throughput, whereas the multistream schedulers achieve a gain around 100 kb during the 100 TTIs of the run, which means 8% of gain. As one can see, because the multistream PF takes the channel conditions into account to select users, it has slightly better performance.

Fig. 2 shows that with doubled queue rates, throughput disparities appear between single user schedulers. Round Robin lead to the lowest throughput, whereas Max Weight and Max Throughput have the highest throughput among the traditional schedulers. As far as multistream scheduling is concerned, one can see that it leads to the best throughputs, even better than the Max Throughput, but no difference really appears between Proportional Fair and

Round Robin strategies. The gain on the traditional Proportional Fair scheduler is 8%, and there is still around 6% gain on Max Throughput. And since our multistream scheduler is based on the Proportional Fair strategy, we can state that it is also fairer than Max Throughput.

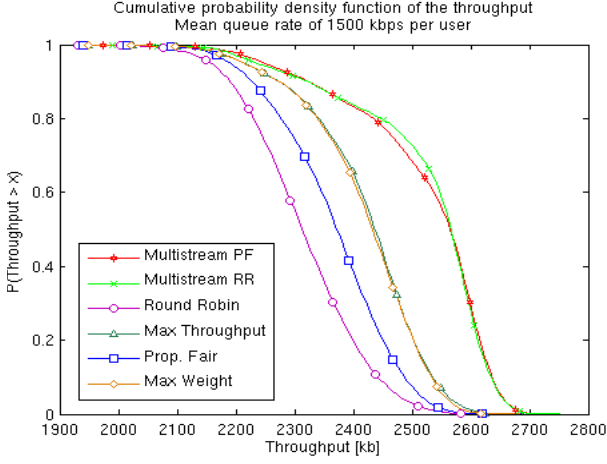


Fig. 2 : Cumulative probability density function of the cell throughput, for 8 users with a mean queue rate of 1500 kbps each.

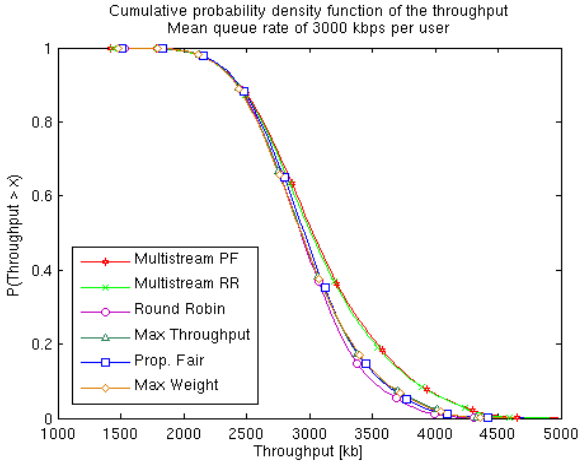


Fig. 3 : Cumulative probability density function of the cell throughput, for 8 users with a mean queue rate of 3000 kbps each.

Finally, as Fig. 3 shows, when queue rates are really large, every user has sufficient data in his/her queue to plainly benefit from all codes. Therefore all schedulers lead to the same throughput, limited by the cell capacity. Indeed, as Figs. 4 & 5 show, with the highest queue rate, most of the time only one user is scheduled within a TTI, such that multistream scheduling is pointless. On the other hand, lower queue rates lead to multiple simultaneous users, up to 8 90% of the time for the smallest queue rate. Indeed, multiple user scheduling is only preferred when a single user can not benefit alone from all radio resources.

To compare the distribution of the number of simultaneous users of the two multistream schedulers, one can say from Figs. 4 & 5 that they behave the same at low and high queue rates. A difference only appears at moderate queue rates, where the multistream Proportional Fair schedules a single user 50% of the time. The multistream Round Robin selects more often multiple users, as it is shown on Fig. 5.

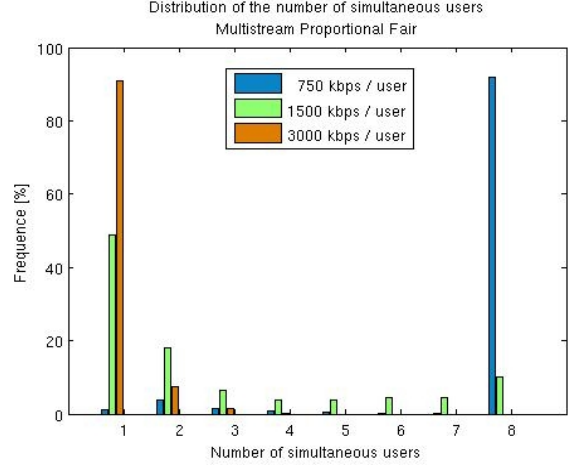


Fig. 4 : Distribution of the number of simultaneous users, for the different queue rates with the multistream PF scheduler

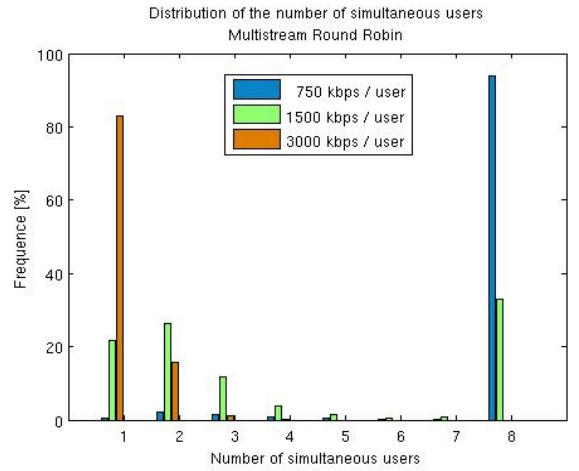


Fig. 5 : Distribution of the number of simultaneous users, for the different queue rates with the multistream RR scheduler

We also tried to double the number of users, each with half of the studied queue rates. The behaviour and the gains are similar: high queue rates lead to no multistream gain whereas low queue rates lead to 6-8% of throughput gain, because low queue rates lead to multiple simultaneous users (15 users 90% of the time with in average 375 kbps for each user queue rate).

As far as fairness is concerned, we evaluated the Jain Index [9]. The advantage of the Jain index is that it is bounded in between $1/N_{Queued}$ and 1, dimensionless and independent of the number of users. If we write \tilde{B}_k the actual transferred throughput of the user k at the end of the simulation, the Jain Index is defined by

$$Jain\ Index \equiv \frac{\left(\sum_{k=1}^{N_{Queued}} \tilde{B}_k \right)^2}{N_{Queued} \cdot \sum_{k=1}^{N_{Queued}} (\tilde{B}_k)^2} . \quad (15)$$

We evaluated the Jain Index of the different schedulers and averaged the results over 2,000 runs of 25 TTIs each. Only 25 TTIs have been considered to avoid long term averaging, where fairness is actually determined by the same queue rate of each user.

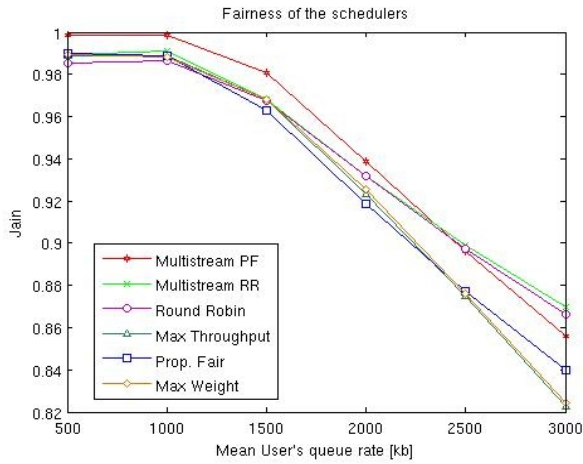


Fig. 6 : Fairness of the different schedulers, depending on the user's queue rate

As one can see on Fig. 6, at low queue rates, all schedulers are as fair as each other, except the multistream Proportional Fair, which is slightly fairer. At moderate queue rates, multistream Proportional Fair becomes as fair as single user Round Robin, but with better throughput. Finally, at high queue rates, multistream Proportional Fair becomes a little less fair than single user Round Robin, but still fairer than single stream Proportional Fair. Max Throughput and Max Weight, which produce the same throughput, lead to equal fairness. As far as Round Robin is concerned, multistream does not seem to really improve fairness, even if the throughput gain is obvious. A strange behavior is that multistream Proportional Fair has better fairness performance than multistream Round Robin for low to moderate queue rates, while they have quite the same cell throughput. This probably comes from the Proportional Fair criteria, which takes into account the queue length and the channel capacity of each user. As a result, users are selected in order to have similar throughput with the multistream Proportional Fair, whereas multistream Round Robin is only fair on the number of TTIs each user has been scheduled.

We also evaluated the fairness after only 10 TTIs, and schedulers behave similarly.

Finally, another advantage of multistream scheduling is that all users can transmit more often than with single stream schedulers, so their throughput is smoother, less bursty. Its effect can be a real advantage for 3G multimedia services, like video streaming for example.

V. CONCLUSIONS

Since scheduling a single user within a TTI can lead to radio resource waste, we introduced a multistream

scheduler. For a good trade-off between fairness among users and throughput, we based our work on a Proportional Fair-like criteria, but actually the criteria can be changed for any other one. The scheduling algorithm has been set up to minimize interference between simultaneous users. We focused on HSDPA transmissions, and evaluate both throughput and fairness performance of the multistream scheduler via computer simulations. Those performance have been compared with four traditional single user schedulers, namely Round Robin, Max Throughput, Proportional Fair and Max Weight, and both throughput and fairness gain has been observed, particularly for low to moderate queue rates. Indeed, when queue rates are too high, a single user can benefit from all radio resource alone, and therefore most of the time the multistream scheduler only select one user per TTI.

For future work, we should also consider Multiple-Input Multiple-Output (MIMO) transmissions, and deal with antenna sharing. MIMO techniques are really promising, and would be able to make transmissions more reliable [10], to bring better throughput [11], and to restrict the antenna emissions to limited directions [12]. Because our throughputs are only estimated by the Shannon bound, it would also be interesting to consider the effects of a SNIR depending coding scheme.

REFERENCES

- [1] R. Kwan, P. Chong, E. Poutiainen et M. Rinne, The Effect of Code-Multiplexing on the HSDPA in a WCDMA Network, in Proceedings of IEEE WCNC'03 conference, 2003
- [2] L. T. Berger, Performance of Multi-Antenna Enhanced HSDPA, PhD Thesis, Aalborg, 2005
- [3] 3GPP TR 25.814 v7.1.0, Physical layer aspects for evolved Universal Terrestrial Radio Access, September 2006
- [4] C. E. Shannon, A Mathematical Theory of Communication, in Bell Sys. Tech. Journal, 27, 1948
- [5] S. Vangipuram et S. Bhashyam, Multiuser Scheduling and Power Sharing for CDMA Packet Data Systems, in Proceedings of IEEE NCC Conference, 2007
- [6] N. Mehta, A. Molisch et L. Greenstein, Orthogonality Factor in WCDMA Downlinks in Urban Macrocellular Environments, in Proceedings of IEEE Globecom 2005 Conference, 2005
- [7] 6th Framework Programme, Information Society Technologies, Wireless World Initiative New Radio (WINNER), IST-2003-507591, <https://www.ist-winner.org>
- [8] Holma and A. Toskala, WCDMA For UMTS - HSDPA Evolutions and LTE, 2007
- [9] R. Jain, D.M. Chiu, and W. Hawe, A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems. DEC Research Report TR-301. 1984
- [10] H. Schulze et C. Lüders, Theory and application of OFDM and CDMA, in John Wiley & Sons, 2005
- [11] I. E. Telatar, Capacity of multi-antenna gaussian channels, in European Transactions on Telecommunications, vol. 10, pp. 585–595, Nov./Dec. 1999
- [12] P. Stoica et R. Moses, Spectral Analysis of Signal, in Pearson Prentice Hall, 2005